

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-305906

(43)Date of publication of application : 02.11.2000

(51)Int.Cl.

G06F 15/16
G06F 12/00
G06F 15/177

(21)Application number : 11-111822

(71)Applicant : HITACHI LTD

(22)Date of filing : 20.04.1999

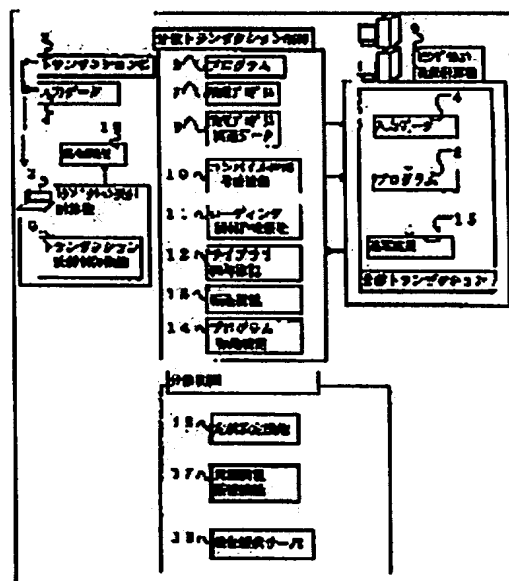
(72)Inventor : SAKURAI NAOKI

(54) DECENTRALIZED TRANSACTION CONTROLLER

(57)Abstract:

PROBLEM TO BE SOLVED: To eliminate the need for the batch updating operation of programs of multiple computers and to facilitate decentralized transaction control by converting and distributing a transaction program together with a precondition program when the transaction is received, and running the program and obtaining its process result.

SOLUTION: When a transaction receiving computer 2 receives a transaction name 3 as a decentralized transaction request and input data 4, a transaction reception control function 5 searches for the program 6 corresponding to the received transaction and a compilation information generating function 10 converts the program into an operable format by a transaction operation computer 9. Then a transfer function 13 transfers the program to a transaction operating computer 9 together with the precondition program 7 and precondition program relative data 8 and a program operating function 14 sets input data 4 in the program 6 and runs the program. The process result 15 is sent back to the transaction reception control function 5.



LEGAL STATUS

[Date of request for examination]

19.03.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-305906

(P2000-305906A)

(43) 公開日 平成12年11月2日 (2000.11.2)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F 15/16	6 2 0	G 0 6 F 15/16	6 2 0 F 5 B 0 4 5
12/00	5 1 8	12/00	5 1 8 A 5 B 0 8 2
15/177	6 7 0	15/177	6 7 0 B

審査請求 未請求 請求項の数 2 O L (全 14 頁)

(21) 出願番号 特願平11-111822

(22) 出願日 平成11年4月20日 (1999.4.20)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 桜井 直樹

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア事業部内

(74) 代理人 100075096

弁理士 作田 康夫

Fターム(参考) 5B045 BB47 GG01 GG04 HH02

5B082 GA02 HA05

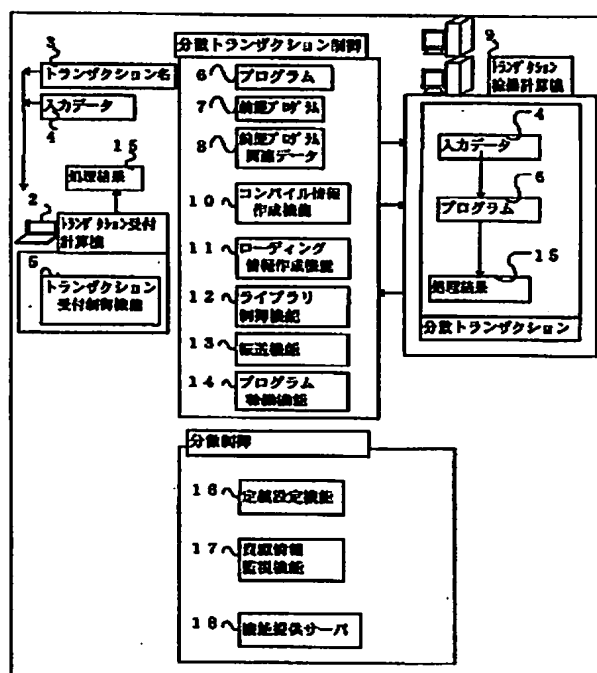
(54) 【発明の名称】 分散トランザクション制御装置

(57) 【要約】

【課題】選んだ計算機のCPUアーキテクチャに合わせプログラムを変換し、前提プログラムと共に転送を行いプログラムを稼働させ演算結果を得る事を可能とし、分散トランザクション制御の容易化する。

【解決手段】計算機割当手段として、トランザクション稼動前に、計算機毎の利用最大負荷とトランザクション毎の利用負荷を定義し、トランザクション要求発生時には利用計算機の利用中負が最も余裕のある計算機を割当てる手段、現在の計算機負荷状況の配布回収手段、利用計算機を個別定義とグループ定義する手段を設ける。

図1



【 特許請求の範囲】

【 請求項1 】 分散トランザクションを利用する際、プログラムをCPUアーキテクチャに合わせて変換し前提プログラムと共に配布する環境作成作業と、トランザクションにデータを与え稼働させ演算結果を得るトランザクション処理を、トランザクション要求受け付け後に一連の流れで自動処理できる事により、分散トランザクション制御を容易にする装置であって、トランザクション要求を受け付ける手段と、該当トランザクションプログラムを稼働するCPUアーキテクチャに合わせてソースプログラムを変換するコンパイル手段と、プログラム及び入力データのローディング情報を作成する手段と、変換したプログラムと前提プログラムを保存するライブラリによりプログラムの再利用と再配布を行う手段と、変換したプログラム及びデータとプログラムの前提プログラムと演算結果を計算機に転送する手段と、プログラムを稼働する手段を具備し、トランザクションを受付けると該当トランザクションプログラムを変換し前提プログラムと共に配布し稼働させ処理結果を得る事を特徴とする分散トランザクション制御装置。

【 請求項2 】 請求項1 記載した分散トランザクションの管理機能となる手段に対し、グループ割り当てした複数の計算機から最適な計算機を自動的に割当てる分散制御を行う装置であって、定義情報に計算機毎の利用最大負荷とトランザクション毎の利用負荷を定義する手段と、現在の利用中負荷の状況を示す資源情報を収集配布する手段と、資源情報からグループ割当てした複数の計算機から稼働負荷の空き状況が最大である計算機を探し出し割当てる手段を持ち、請求項1 記載の機能として利用する計算機を自動割当管理する事を特徴とする分散トランザクション制御装置。

【 発明の詳細な説明】

【 0001 】

【 発明の属する技術分野】 本発明は、複数計算機を使用した分散トランザクション処理に関わり、特に分散トランザクションプログラムの稼働制御及び分散トランザクション管理機能の分散制御に関する。

【 0002 】

【 従来の技術】 従来の分散トランザクション処理手法としては、複数の計算機のトランザクション機能を連携利用する事により、計算機の負荷を分散する手法が一般に知られている。分散トランザクション処理手法は、複数の計算機のトランザクション機能を連携利用する事により、計算機の負荷を分散するものである。この手法では、トランザクション稼働前に、前提プログラムを用意し複数の計算機のCPUアーキテクチャに合わせてプログラムを変換させ分散処理機能を持つ計算機に転送する環境作成準備作業が必須である。

【 0003 】 特開平5-173988号公報(分散処理方式及び該分散処理に適用されるトランザクション処理

方式)では、それぞれの計算機のサービス管理部は、クライアント・プロセスの要求サービスの問い合わせに対してどのサーバ・プロセスが実行可能か判断し該当するサーバ・プロセスに対し前記通信処理部を用いてサービスを要求することを特徴としているが、実行可能状態である事が前提であり、プログラムを事前に計算機に転送する事が必須である。

【 0004 】 一方、プログラムの配布を行う手法では、定義した時刻にバージョンアップしたプログラムを配布しアプリケーションとして登録する手法である。特開平8-179936号公報(プログラム管理システム)では、プログラムを転送しインストールを行いバージョン管理はするが、トランザクションを受付た後でトランザクション処理を行う機能を持たない。

【 0005 】 たとえば、分散トランザクションプログラムをバージョンアップする場合には、プログラムを変換し稼働計算機に転送後に稼働させ稼働結果を得る一連の作業をトランザクション要求を元に作動する事ができない為、分散トランザクション稼働予定の計算機全てに対してトランザクションを未稼働の状態にしてバージョンアップしたプログラムを一括配布し、定義した時刻に稼働する必要がある、複数計算機に対する多大な注意と管理工数がかかる為、分散トランザクションの制御手法の容易化が求められていた。

【 0006 】

【 発明が解決しようとする課題】 上記従来方式である、分散トランザクションの手法では複数計算機に対し一括してCPUアーキテクチャ毎に、分散トランザクションプログラムを配布し稼働する稼働環境作成作業が必要であり、利用者に対し大変な負担を残していた。

【 0007 】 本発明の目的は、分散トランザクション処理をする計算機に対し稼働環境作成作業をする事なく、トランザクション要求を受け付けた時点で分散トランザクションプログラムを稼働する稼働負荷の空き状況が最大である計算機を選び、選んだ計算機のCPUアーキテクチャに合わせてプログラムを変換し前提プログラムと共に、転送を行う稼働環境作成作業を行い、プログラムを稼働させトランザクション処理を行い演算結果を得る事ができる事により、計算機の利用負荷に応じて別の計算機に対しプログラムを転送し稼働結果を得る事が可能となり複数計算機のプログラムの一括更新作業が不要にし、分散トランザクション制御を容易にする事を目的としている。

【 0008 】

【 課題を解決するための手段】 上記目標を達成する為には、ユーザは、トランザクションプログラム毎に、プログラム変換方法と利用計算機をグループ指定する。トランザクション要求を受け付けた時点で、トランザクションを受け付ける構成要素と、受付けたトランザクションに対するプログラム変換方法と利用計算機のグループを

探し、探し出した計算機のグループから分散トランザクションプログラムを稼働する稼働負荷の空き状況が最大である計算機を選び出し計算機を割り当てる構成要素と、選んだ計算機のCPUアーキテクチャに合わせてプログラムを変換する構成要素と、前提プログラムと共に転送する構成要素とプログラムを稼働する構成要素と、演算結果を転送する構成要素により演算結果を得る事が可能となり、容易に分散トランザクション環境を構築する事が可能となった。指定計算機グループから選択割当てる為、分散トランザクション環境の構築変更が容易になった。

【0009】

【発明の実施の形態】以下本発明による分散トランザクション制御装置の実施例を図面を参照しつつ説明する。

【0010】(1) 分散トランザクション制御装置概要
図1に本発明による分散トランザクション制御装置1の実施例のシステム構成図を示す。分散トランザクション制御装置1は、トランザクション受付計算機2で、分散トランザクション要求となるトランザクション名3及び入力データ4を受付けると、トランザクション受付制御機能5により受け付けたトランザクションに該当するプログラム6を探し、コンパイル情報作成機能10によりトランザクション稼働計算機9で稼働可能形式に変換し、ローディング情報作成機能11によりプログラム6と入力データ4のローディング情報を作成し、ライブラリ制御機能12により再変換防止の為に変換情報を保存し、前提プログラム7と前提プログラム関連データ8と共に転送機能13によりトランザクション稼働計算機9に転送しプログラム稼働機能14によりプログラム6に入力データ4を設定し稼働させ、処理結果15をトランザクション受付制御機能5に返信し、プログラムに処理結果を返す。

【0011】上記したトランザクション制御を行うコンパイル作成情報機能10とローディング情報作成機能11とライブラリ制御機能12とプログラム稼働機能14の分散制御の為、定義設定機能16により稼働場所をグループ定義し、資源情報監視機能17により現在の負荷情報を管理し、トランザクション受付制御機能5により最も余裕のある計算機をトランザクション稼働場所に割当てる。トランザクション受付制御機能5を持つトランザクション受付計算機2とプログラム稼働機能14を持つトランザクション稼働計算機9の負荷上昇防止の為、プログラム変換を行うコンパイル情報作成機能10とライブラリ情報作成機能11を稼働する機能提供サーバ18を用意する。

【0012】分散トランザクション制御装置1は、分散トランザクション制御するトランザクション受付制御機能5 コンパイル情報作成機能10 ローディング情報作成機能11 ライブラリ制御機能12 転送機能13 プログラム稼働機能14と、前記機能を分散制御する定義

設定機能16 資源情報監視機能17 から構成されている。

【0013】受け付けたトランザクション名3から、分散トランザクション制御装置1の各機能を行う計算機を割りあてるトランザクション受付制御機能5を図2で説明し、図3でソースから実行形式を作成するコンパイル情報作成機能10とローディング情報作成機能11の制御方式の方式を説明し、図4及び図5では作成した情報を保存するライブラリ制御機能12の制御方式を説明する。図6では分散トランザクション制御処理の流れを説明する。図7と図8では、コンパイル情報作成機能10とローディング情報作成機能11とプログラム稼働機能12を分散制御する方式を説明し、図9では、負荷監視情報による計算機割り当て指定方式を説明する。

【0014】(2) 分散トランザクション制御方式

(a) トランザクション受付制御

分散トランザクション制御装置1は、トランザクションを受け付けると、トランザクション受付制御機能5により、トランザクション名に対応するプログラムを探す。

【0015】トランザクション受付開始前に利用者は、トランザクション名に対応するプログラム情報と利用計算機情報をトランザクション定義情報19及びプログラム変換方法22に定義する。次に、トランザクション名20に対応するプログラム名称21とプログラム変換方法22を定義する。

【0016】プログラム変換方法22として、コンパイル情報作成機能10に渡す情報をコンパイルパラメータ23に定義し、ローディング情報の作成方法としてローディング情報作成機能11に渡す情報をローディングパラメータ24に定義し、プログラムの稼働情報としてプログラム稼働機能14に渡す情報を稼働パラメータ25に定義し、コンパイル情報作成機能10とローディング情報作成機能11とプログラム稼働機能14の稼働計算機情報を機能分散制御情報26に定義し、ソースとコンパイル変換後情報と実行形式のプログラム保管場所をライブラリ制御情報29に定義する。

【0017】トランザクション受付制御機能5は、受け付けたトランザクション名3に対し上記した定義情報をもとに、対応するプログラム名称21を探し、プログラム変換方法と利用計算機情報を取得し、コンパイル情報作成機能10とローディング情報作成機能11プログラム稼働機能14の稼働場所とパラメータを決定し、変換プログラム情報の保存場所を決定し、各種機能に処理を移す。

【0018】図2の例は、トランザクション名が「期末決算処理」のプログラム変換方法を示し、トランザクション処理を行うプログラム名称21「KIMATSU」を作成する為に、ライブラリ制御情報29に指定されたソースを、コンパイルパラメータ23とローディングパラメータ24をODBC(=Open Data base

se Connectivity: DBの操作機能であり本文ではDB操作情報に変換する意味で用いる。以後(ODBCと記す)を設定し、稼働パラメータ25に優先順序1を設定し、前提プログラム情報27として「PRG-A」を設定し、前提プログラム関連情報28として「DB定義」を設定し、作成したコンパイル変換後情報と実行形式の保存場所をライブラリ制御情報29に設定する例を示す。

【0019】(b)プログラム6と入力データ4の変換方式

図3ではコンパイル情報作成機能10とローディング情報作成機能11の動作を説明する。

【0020】コンパイル情報作成機能10は、プログラムを作成する際のコンパイラ+リンカ33の機能でありアーキテクチャの異なる計算機に合わせて変換する機能を持ち、トランザクション受付制御機能5に指定されたソース形式のプログラム30を、トランザクション定義情報19に指定されたコンパイルパラメータ23を設定しコンパイル変換後情報31に変換する。

【0021】ローディング情報作成機能11は、プログラム実行時にロード23が行うメモリ配置情報を作成する機能でありCPUアーキテクチャ毎に異なる情報を作成でき、コンパイル変換後情報31と入力データ4をトランザクション定義情報19に指定されたローディングパラメータ24を設定しローディング情報32に変換する。

【0022】(c)ライブラリ制御機能

図4では、ライブラリ制御機能12と、保存再利用する情報の種別と目的を説明する。ライブラリ制御機能12は、複数回の分散トランザクション起動要求発生時にプログラム再変換防止の為、ライブラリ制御情報29の指定場所のライブラリ制御機能12に変換情報を保存35し、再利用36する。ライブラリ制御情報29に、プログラム関連としてソース形式のプログラム30コンパイル変換後情報31ローディング情報32を保存し、トランザクション稼働環境として前提プログラム7前提プログラム関連データ8を保存する。

【0023】ソース形式のプログラム30は、プログラム作成の為保存する。コンパイル変換後情報31は、異なるOSでもコンパイルパラメータが同一時に再利用する為保存する。ローディング情報32は、プログラムを再稼働又は他計算機に転送後稼働する為保存する。前提プログラム7および前提プログラム関連データ8は、分散トランザクションの環境構築の為保存する。

【0024】図5では、ライブラリ制御情報29を説明する。プログラム情報の保存35及び再利用36時の制御情報として使用し、プログラム情報を保存35時、ライブラリ制御情報29に対しプログラム名称21を設定し、保存するプログラムの変換レベルをプログラム変換レベル37に指定し、変換パラメータ情報としてコンパ

イルパラメータ23とローディングパラメータ24を設定し、保存場所38と保存ファイル名39を設定し保存35する。

【0025】図4の例は、「KIMATSU」プログラムをコンパイルパラメータ23とローディングパラメータ24をODBCに設定し作成したコンパイル変換後情報31を計算機AのC:\¥LIBAにファイル名「KESSAN1.OBJ」で保存した設定情報例を示す。

【0026】(d)分散トランザクション制御全般の流れ

図6では、分散トランザクション制御全般の流れを説明する。トランザクション受付制御機能5を持つトランザクション受付計算機2が、分散トランザクション要求としてトランザクション名3入力データ4を受け、トランザクション名3から該当プログラム名称21を検索し、コンパイル作成情報機能10とローディング情報作成機能11とプログラム稼働機能14の稼働計算機を割当て、処理を要求する。コンパイル情報作成機能10はソース形式のプログラム30をコンパイル変換後情報31に変換し、ローディング情報作成機能11は入力データ4とコンパイル変換後情報31からローディング情報32を作成する。

【0027】ライブラリ制御機能12は、コンパイル変換後情報31及びローディング情報32を、トランザクション要求再発生時の再変換負荷削減の為保存35する。転送機能12は、コンパイル変換後情報31及びローディング情報32と共に、トランザクション稼働環境として前提プログラム7と前提プログラム関連データ8を、トランザクション稼働計算機5に転送する。

【0028】プログラム稼働機能14は、前提プログラム7を稼働させ、ローディング情報32に変換したプログラムに入力データをパラメータに設定し稼働させ、処理結果15を得る。転送機能12は、処理結果15をトランザクション受付制御機能5に返信し、処理結果15をプログラムに戻しトランザクション処理を完了する。

【0029】(3)計算機割り当て制御

分散トランザクション制御装置1は、コンパイル情報作成機能10とローディング情報作成機能11とライブラリ情報作成機能12とプログラム稼働機能14の計算機割り当て時に、固定割り当て方法とグループ割り当て方式を持つ。図7では固定割り当て方法を説明し、図8ではグループ割り当て方法を説明し、図9では計算機の稼働負荷情報及び配布通知方法と計算機割り当て方法を説明する。

【0030】(a)計算機固定割り当て制御

図7では、前記した分散トランザクション制御装置1のコンパイル情報作成機能10とローディング情報作成機能11とプログラム稼働機能14の稼働計算機を固定割り当てる為、トランザクション定義情報19の機能分散制御情報26の定義内容と制御方式を説明する。

【0031】機能分散制御情報26に、コンパイル情報

10

20

30

40

50

作成機能10の稼働計算機をコンパイル情報作成機能稼働場所40に指定し、ローディング情報作成機能11の稼働計算機をローディング情報作成機能稼働場所41に指定し、プログラム稼働機能14の稼働場所を分散ランザクション稼働場所42に指定する。作成したコンパイル変換後情報31とローディング情報32の保存場所は、機能分散制御情報26のライブラリ制御情報29に指定する。

【0032】高負荷になるランザクション受付制御機能5を持つランザクション受付計算機2とプログラム稼働機能を持つランザクション稼働計算機5以外に負荷分散する為にプログラム変換のみを行う機能提供サーバ18として利用する事が可能である。図6の例では、ODBCが必須であるランザクション処理を行う計算機として、ランザクション受付を計算機Aで行い、ランザクション稼働を計算機Bで行い、ランザクション処理の負担軽減の為プログラム変換作業を行う機能提供サーバとして計算機Cを割り当てた例を示す。

【0033】(b) 計算機グループ割り当て制御

図8では、コンパイル情報作成機能10とローディング情報作成機能11とプログラム稼働機能14を計算機グループから割り当てる。ランザクション受付制御機能5と機能分散制御情報26への設定内容と制御方法を説明する。

【0034】機能分散制御情報26に、コンパイル情報作成機能10とローディング情報作成機能11とプログラム稼働機能14の稼働場所をグループ指定する為、分散ランザクション稼働グループ43を指定する。分散ランザクション稼働グループ43には、グループ識別子として稼働グループ名44を指定し、コンパイル情報作成機能10の稼働計算機グループをコンパイル情報作成機能グループ45に指定し、ローディング情報作成機能11の稼働計算機グループをローディング情報作成グループ46に指定し、プログラム稼働機能12の稼働計算機グループをプログラム稼働グループ47に指定する。

【0035】計算機のグループ情報として計算機グループ情報48には計算機グループ名49を指定し、計算機のアドレス情報など計算機管理情報50に設定する。作成したコンパイル変換後情報31とローディング情報32の保存場所は、機能毎の計算機グループ情報48のライブラリ制御情報29に指定する。

【0036】ランザクション受付制御機能5は、受け付けたランザクション名3から、該当するプログラムの変換方法22及び稼働計算機情報を機能分散制御情報26から見つけ、プログラム稼働機能グループ47に指定された複数の計算機から利用負荷の余裕が最大の計算機をプログラム稼働機能14として割り当て、保存場所を示すライブラリ制御情報29を元にコンパイル情報作成機能稼働グループ45からコンパイル情報作成機能10

を割り当てローディング情報作成機能稼働グループ46からローディング情報作成機能11を割り当てる。図7の例は、プログラム変換するタイプが複数ある使用例を示す。

【0037】(c) 資源情報による負荷監視と計算機割り当て方式

図9を用いて、計算機の利用負荷状況を示す資源情報51と、資源情報51の収集配布を行う資源情報監視機能17を説明する。利用者は、分散ランザクションの稼働負荷値として計算機割当機能18を持つ計算機のプログラム変換方法22にプログラム利用負荷54を設定し、計算機側の情報として計算機名称55と利用上限として最大利用負荷56を設定する。

【0038】ランザクションの稼働中は、稼働中利用負荷合計57に、稼働しているランザクションのプログラム利用負荷54の合計値を設定し、稼働分散中ランザクション数59を設定する。最大利用負荷56から稼働中利用負荷合計57を減算した値が計算機の余裕度を示す。ランザクション受付制御機能5は、計算機の余裕度が最大の計算機をランザクション稼働計算機9に割り当てる。差が同じ場合は、稼働分散中ランザクション数58の少ない計算機を負荷が低い計算機と判断する。

【0039】資源情報52の配布手段は、ランザクション稼働計算機9から計算機割当機能18に通知52する方法。計算機割当機能18からランザクション稼働計算機9を調査53する方法。機能提供サーバ18で一括管理する方法がある。資源情報51は、ランザクション稼働計算機9の利用負荷状況としてランザクション受付制御機能5に配布されランザクション稼働計算機9を決定基準となる。

【0040】(4) 処理の流れ

次に処理の流れを、説明する。

【0041】(a) 処理概要

図10では、処理概要を示す。分散ランザクション要求を受け付ける前に、計算機の利用の仕方とランザクションの稼働方法を定義設定機能16を用い定義する。定義設定機能16詳細は図11にて説明する(1001)。プログラム情報としてプログラム変換方法22のライブラリ制御情報29にソースを保存する。(1002)。作業終了後に、分散ランザクション要求受け付けを開始する(1003)。

【0042】ランザクション受付制御機能5は、受け付けたランザクション名3を定義しているランザクション定義情報19を探す。次にプログラム変換方法22から余裕度が最大であるランザクション稼働計算機を探し、プログラム変換方法を決定し、コンパイル情報作成機能10とローディング情報作成機能11とプログラム稼働機能14の計算機の稼働場所を割り当てる。

【0043】ランザクション受付制御機能5の詳細は

図12にて説明する(1004)。コンパイル情報作成機能10の指定が有る場合コンパイル情報作成機能10にてコンパイル変換後情報31を作成する。コンパイル情報作成機能13の処理詳細は図13にて説明する(1005)。ローディング情報作成機能11指定時は、ローディング情報作成機能11でローディング情報32を作成する。

【0044】ローディング情報32の処理詳細は図14にて説明する(1006)。以上の動作でランザクションの稼働情報作成が終了し、プログラム稼働機能14にてプログラムを稼働する(1007)。プログラム処理結果15をランザクション受付制御機能5に転送し(1008)、処理結果15をプログラムに戻し(1009)ランザクション要求を終了する。

【0045】(b)定義設定機能16の処理の流れ
図11では、計算機の利用の仕方とランザクションの稼働方法を定義する定義設定機能16の処理の流れを説明する。

【0046】ランザクション定義情報19として、ランザクション名称20とプログラム名称21を設定する(1101)。次にランザクション定義情報19に全てのプログラム変換方法22を指定する(1102)。プログラム変換方法22にコンパイルパラメータ23とローディングパラメータ24と稼働パラメータ25と前提プログラム情報27と前提プログラム関連情報28とライブラリ制御情報29を設定する(1103)。

【0047】次にプログラム変換方法22に全ての機能分散定義情報26を指定する(1104)。計算機分散利用の為、機能分散定義情報26には、計算機の個別指定とグループ指定がある(1105)。計算機個別指定時、機能分散制御情報26のコンパイル情報作成機能稼働場所40とローディング情報作成機能稼働場所41と分散ランザクション稼働場所42に個別に計算機を指定する(1106)。

【0048】次にコンパイル変換後情報31とローディング情報32保管場所を、機能分散定義情報26のライブラリ制御情報29に設定する(1107)。計算機グループ指定時、機能分散制御情報26の分散ランザクション稼働グループ43を設定する(1108)。

【0049】次に分散ランザクション稼働グループ43に、コンパイル情報作成機能10の稼働計算機のグループ情報をコンパイル情報作成機能グループ45に指定し、ローディング情報作成機能11の稼働計算機のグループ情報をローディング情報作成機能グループ46に指定し、プログラム稼働機能14の稼働計算機のグループ情報をプログラム稼働機能グループ47に指定する(1109)。

【0050】次にグループの個別計算機情報として、計算機グループ情報48に、計算機のアドレス情報を計算

機情報50に設定し、コンパイル変換後情報31とローディング情報32保管場所をライブラリ制御情報29に設定する(1110)。

【0051】(c)ランザクション受付制御機能5の処理の流れ

図12はランザクション受付制御機能5の処理の流れを示した図である。ランザクション受付制御機能5は、受け付けたランザクション要求のランザクション名3を定義しているランザクション定義情報19を探す(1201)。次にソース形式のプログラム30より新しく作成したローディング情報32の記録情報が機能分散制御情報26に設定してあるプログラム変換方法22を全て検索し(1202)、発見時と未発見時とで処理を分ける(1203)。

【0052】新しいローディング情報32の発見時には、ローディング情報32が有るプログラム変換方法22から、ランザクション稼働の余裕度が最大値であるランザクション稼働計算機9を探し、利用するプログラム変換方法22および機能分散制御情報26を決め(1204)、プログラム稼働機能14の計算機を決める(1205)。新しいローディング情報32の未発見時には、受け付けたランザクション名3を定義しているプログラム変換方法22からソース形式のプログラム30より新しく作成したコンパイル変換後情報31の記録情報が機能分散制御情報26に設定してあるプログラム変換方法22を全て探す(1206)。

【0053】次に、新しいコンパイル変換後情報31の発見時と未発見時とで処理を分ける(1207)。新しいコンパイル変換後情報31の発見時には、コンパイル変換後情報31を探せた定義からランザクション稼働の余裕度が最大であるランザクション稼働計算機9を探し利用するプログラム変換方法22および機能分散制御情報26を決め(1208)、プログラム稼働機能14とローディング情報作成機能11の稼働計算機を決める(1209)。

【0054】新しいコンパイル変換後情報31の未発見時は、受け付けたランザクション名3を定義しているプログラム変換方法22からランザクション稼働の余裕度が最大であるランザクション稼働計算機9を探し利用するプログラム変換方法22および機能分散制御情報26を決め(1210)、プログラム稼働機能14とローディング情報作成機能11とコンパイル情報作成機能10の稼働計算機を決める(1211)。以上の処理で今後の処理をする計算機を決める。

【0055】(d)コンパイル情報作成機能

図13にコンパイル情報作成機能の処理の流れを説明する。コンパイル情報作成機能9は、受け付けたコンパイル変換後情報31の作成要求が分散ランザクション稼働グループ48指定時と個別指定時で処理を分ける(1301)。グループ指定時は、コンパイル情報作成機能

グループ45の計算機グループ情報48のライブラリ制御情報29に指定するソース形式のプログラムの保管場所を探す(1302)。個別指定時、機能分散制御情報26のライブラリ制御情報29からソース形式のプログラムの保管場所を探す(1303)。

【0056】次にソース形式のプログラム30の保管場所の自計算機の有無により処理を分ける(1304)。自計算機にソース形式のプログラム30が無い場合は、ソース形式のプログラム30の保管場所のコンパイル情報作成機能10に処理を移す(1305)。自計算機にソース形式のプログラム30が有る場合は、コンパイルパラメータ23を設定してコンパイル変換後情報31を作成し(1306)、コンパイル変換後情報31をライブラリ制御情報29の指定場所に転送機能13にて転送し(1307)、ライブラリ制御機能12に保存し(1308)、ローディング情報作成機能11に処理を移す(1309)。

【0057】(e)ローディング情報作成機能

図14にローディング情報作成機能11の処理の流れを説明する。ローディング情報作成機能11は、受付けたローディング情報32作成要求が分散トランザクション稼働グループ48指定時と個別指定時で処理を分ける(1301)。グループ指定時、ローディング情報作成機能グループ46の計算機グループ情報のライブラリ制御情報29の設定情報であるコンパイル変換後情報31の保管場所を探す(1402)。個別指定時、機能分散制御情報26のライブラリ制御情報29からコンパイル変換後情報31の保管場所を探す(1403)。

【0058】次に、コンパイル変換後情報31保管場所の自計算機の有無で処理を分ける(1404)。自計算機にコンパイル変換後情報31が無い場合、コンパイル変換後情報31保管場所のローディング情報作成機能11に処理を移す(1405)。自計算機にコンパイル変換後情報31が有る場合、ローディングパラメータ24を設定してローディング情報32を作成し(1406)、ローディング情報32をライブラリ制御情報29の指定場所に転送機能13を用いて転送し(1407)、ローディング機能11に保存し(1408)、トランザクション稼働機能に処理を移す(1409)。

【0059】(e)プログラム稼働機能

図15にプログラム稼働機能14の処理の流れを説明する。トランザクション稼働要求を受けたプログラム稼働機能14は、利用中負荷状況を示す稼働中利用負荷合計57に利用負荷54を加算し(1501)、前提プログラム関連データ8を設定し前提プログラム7を起動し(1502)、ライブラリ制御情報29の指定場所からローディング情報を読出し(1503)、稼働パラメータ25と入力データ4を設定し分散トランザクションを稼働し(1504)、トランザクション処理終了後に稼働中利用負荷合計57から利用負荷54を減算し(15

05)、処理結果をトランザクション受付制御機能5に転送し(1506)、処理をトランザクション受付制御機能5に移し終了する(1507)。

【0060】

【発明の効果】従来、分散トランザクション環境を作成する際に、稼働環境の事前作成が必要であり利用者に対し大変な負担を残していた。本発明では、計算機に対し事前にプログラムの稼働環境を整備しなくてもトランザクション要求発生時に、分散トランザクションプログラムを稼働する負荷の空き状況が最大の計算機を選び、選んだ計算機のCPUアーキテクチャに合わせてプログラムを変換し、前提プログラムと共に転送を行いプログラムを稼働させ演算結果を得る事が可能になり、分散トランザクション制御が容易になった。

【図面の簡単な説明】

【図1】分散トランザクション制御装置1の概要を示す図。

【図2】トランザクション受付制御機能5を示す図。

【図3】コンパイル情報作成機能10とローディング情報作成機能11を示す図。

【図4】ライブラリ制御機能12を示す図。

【図5】ライブラリ制御情報29を示す図。

【図6】分散トランザクション制御処理の流れを示す図。

【図7】機能分散制御(計算機個別割り当て指定方式)を示す図。

【図8】機能分散制御(計算機グループ割り当て制御)を示す図。

【図9】資源情報による負荷監視と計算機割り当て方式を示す図。

【図10】分散トランザクション制御装置1の処理の流れを示す図。

【図11】定義設定機能16の処理の流れを示す図。

【図12】トランザクション受付制御機能5の処理の流れを示す図。

【図13】コンパイル情報作成機能10の処理の流れを示す図。

【図14】ローディング情報作成機能11の処理の流れを示す図。

【図15】トランザクション稼働機能14の処理の流れを示す図。

【符号の説明】

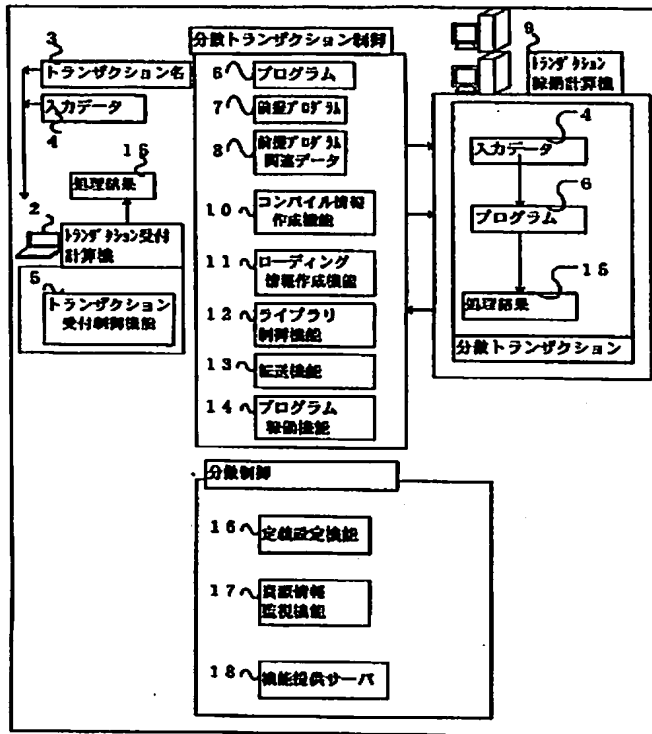
1…分散トランザクション制御装置、2…トランザクション受付計算機、3…トランザクション名、4…入力データ、5…トランザクション受付制御機能、6…プログラム、7…前提プログラム、8…前提プログラム関連データ、9…トランザクション稼働計算機、10…コンパイル情報作成機能、11…ローディング情報作成機能、12…ライブラリ制御機能、13…転送機能、14…プログラム稼働機能、15…処理結果、16…定義設定機

13

能、17…資源情報監視機能、18…機能提供サーバ、
19…トランザクション定義情報、20…トランザク
ション名称、21…プログラム名称、22…プログラム変
換方法、23…コンパイルパラメータ、24…ローディ
ングパラメータ、25…稼働パラメータ、26…機能分
散制御情報、27…前提プログラム情報、28…前提プ
ログラム関連情報、29…ライブラリ制御情報、30…
ソース形式のプログラム、31…コンパイル変換後情
報、32…ローディング情報、33…コンパイラ+リン
カ、34…ローダ、35…保存、36…再利用、37…
プログラム変換レベル、38…保存場所情報、39…保

【 図1 】

図1



14

存ファイル、40…コンパイル情報作成機能稼働場所、
41…ローディング情報作成機能稼働場所、42…分散
トランザクション稼働場所、43…分散トランザクシ
ョン稼働グループ、44…稼働グループ名、45…コンパ
イル情報作成機能グループ、46…ローディング情報作
成機能グループ、47…プログラム稼働機能グループ、4
8…計算機グループ情報、49…計算機グループ名、5
0…計算機情報、51…資源情報、52…通知、53…
調査、54…プログラム利用負荷、55…計算機名称、
56…最大利用負荷、57…稼働中利用負荷合計、58
…稼働中分散トランザクション数。

【 図2 】

図2

19 トランザクション定義情報

#	設定情報	設定例
1	トランザクション名称	期末決算処理
2	プログラム名称	KIMATSU
3	プログラム変換方法1	UNIX変換
4	プログラム変換方法2	PC変換
	:	その他

22 プログラム変換方法

	設定情報	設定例
1	コンパイルパラメータ	: ODBC
2	ローディングパラメータ	: ODBC
3	稼働パラメータ	: 優先順序1
4	機能分散制御情報	: 定義情報名
5	前提プログラム情報	: PRG-A
6	前提プログラム関連情報	: DB定義
7	ライブラリ制御情報 (ソース変換後情報 実行形式/前提プログラム/ 前提プログラム関連データ)	: 定義情報名

【 図5 】

圖 5

```

graph TD
    subgraph Left_Process [ ]
        direction TB
        S30_1[ソース形式のプログラム 30] --> C33_1[コンパイラ + リンカ 33]
        C33_1 --> A31_1[コンパイル変換後情報  
(アセンブラレベル) 31]
        A31_1 --> L34_1[ローダ 34]
        L34_1 --> R32_1[ローディング情報  
(ローディング可能レベル) 32]
    end

    subgraph Right_Process [ ]
        direction TB
        S30_2[ソース形式のプログラム 30] --> C23_2[コンパイルパラメータ 23]
        C23_2 --> F10_2[コンパイル情報作成機能 10]
        F10_2 --> A31_2[コンパイル変換後情報  
(アセンブラレベル) 31]
        A31_2 --> L2_2[ローディングパラメータ 2]
        ID4_2[入力データ 4] --> L2_2
        L2_2 --> F11_2[ローディング情報作成機能 11]
        F11_2 --> R32_2[ローディング情報  
(ローディング可能レベル)]
    end

```

#	設定情報	設定例
1	プログラム名称	KIMATSU
2	プログラム実行レベル 0 : ソース形式のプログラム 1 : コンパイル実行後情報 2 : ローディング情報 3 : 前段プログラム 4 : 前段プログラム関連データ	1 : コンパイル実行後情報
3	コンパイルパラメータ情報	: ODBC
4	ローディングパラメータ情報	: ODBC
5	保存場所情報	: 計算機A.C.: WLIBA
6	保存ファイル	作成日 : 98/7/1 作成 ファイル名 : 600KB ファイル名 : KESSAN1.job

48

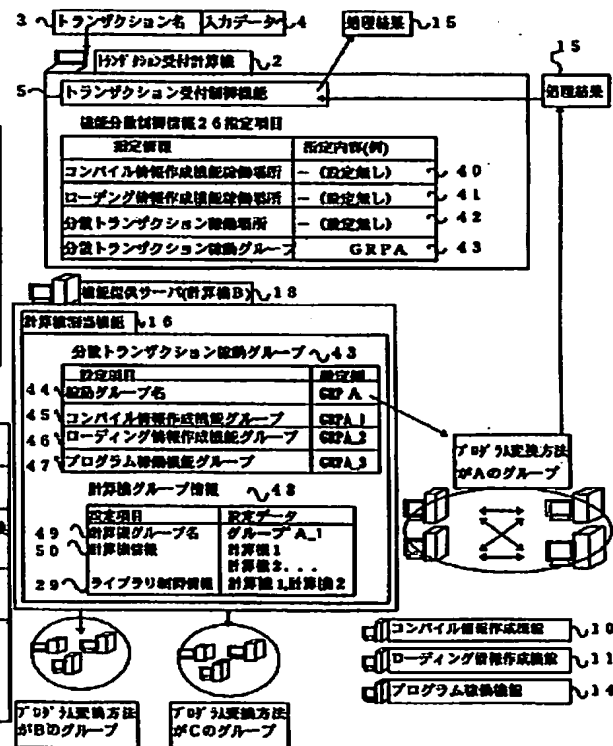
圖4

```

graph LR
    30[30 ソース形式のプログラム] --> 10[10 コンパイル情報作成機能]
    10 --> 31[31 コンパイル変換情報]
    31 --> 11[11 ローディング情報作成機能]
    11 --> 32[32 ローディング情報]
    30 --> 35[35]
    31 --> 35
    32 --> 35
    35 --> 36[36 保存]
    35 --> 36[36 再利用]
    36 --> 29[29 ライブラリ  
制御情報]
  
```

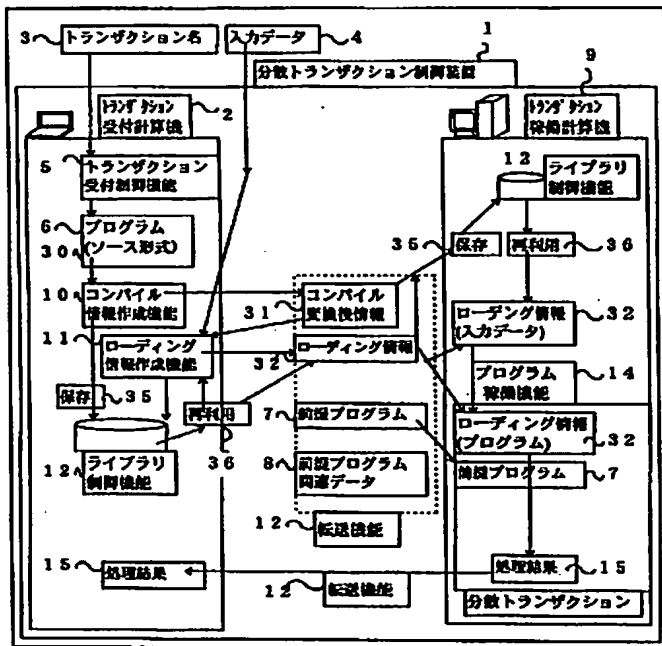
Figure 1 is a block diagram of a program processing system. It shows a flow from 'Source Program' (30) to 'Compile Information Creation Function' (10), then to 'Compile Information Conversion Function' (31), then to 'Loading Information Creation Function' (11), and finally to 'Loading Information' (32). These functions are connected to a central processing unit (35) which also handles 'Storage' (36) and 'Reuse' (36). The system also includes a 'Library Creation Information' (29) block.

#	情報形式	目的
30	1 ソース形式のプログラム	プログラムを作成する為
31	2 コンパイル変換後情報	OSが異なる場合でも再度コンパイル変換しなくても良い
32	3 ローディング情報	プログラムを再稼働又は仮送込稼働する為
7	4 前送プログラム	分散トランザクション検査実行後情報の為
8	5 前送プログラム関連データ	



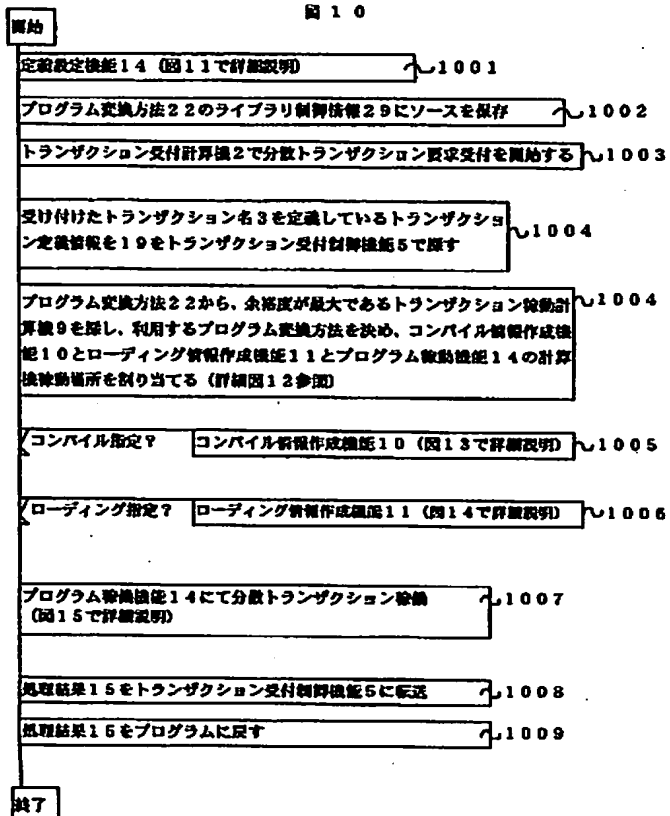
【 図6 】

図6



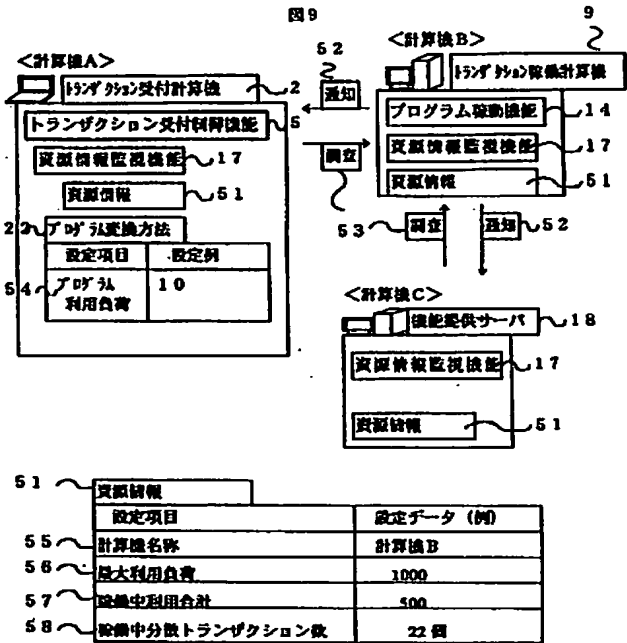
【 図10 】

図 10



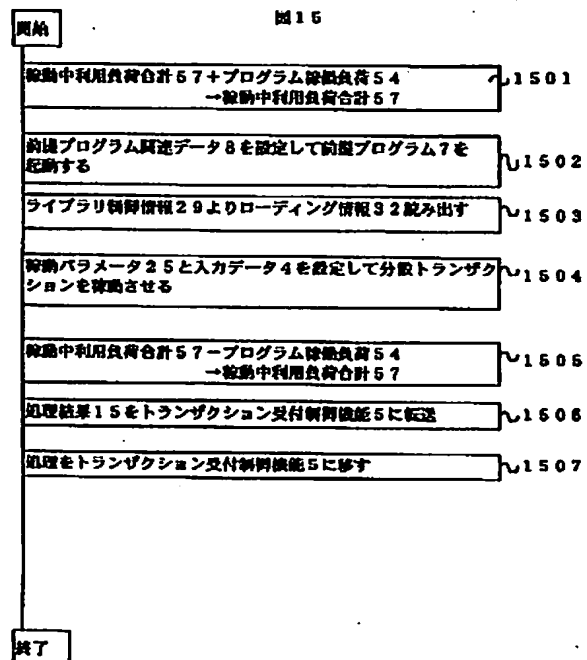
【 図9 】

図9

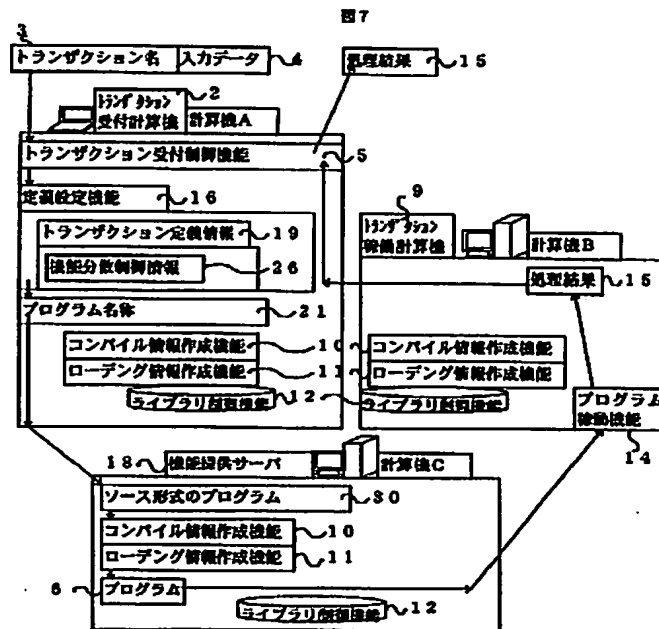


【 図15 】

図15



【 図7 】

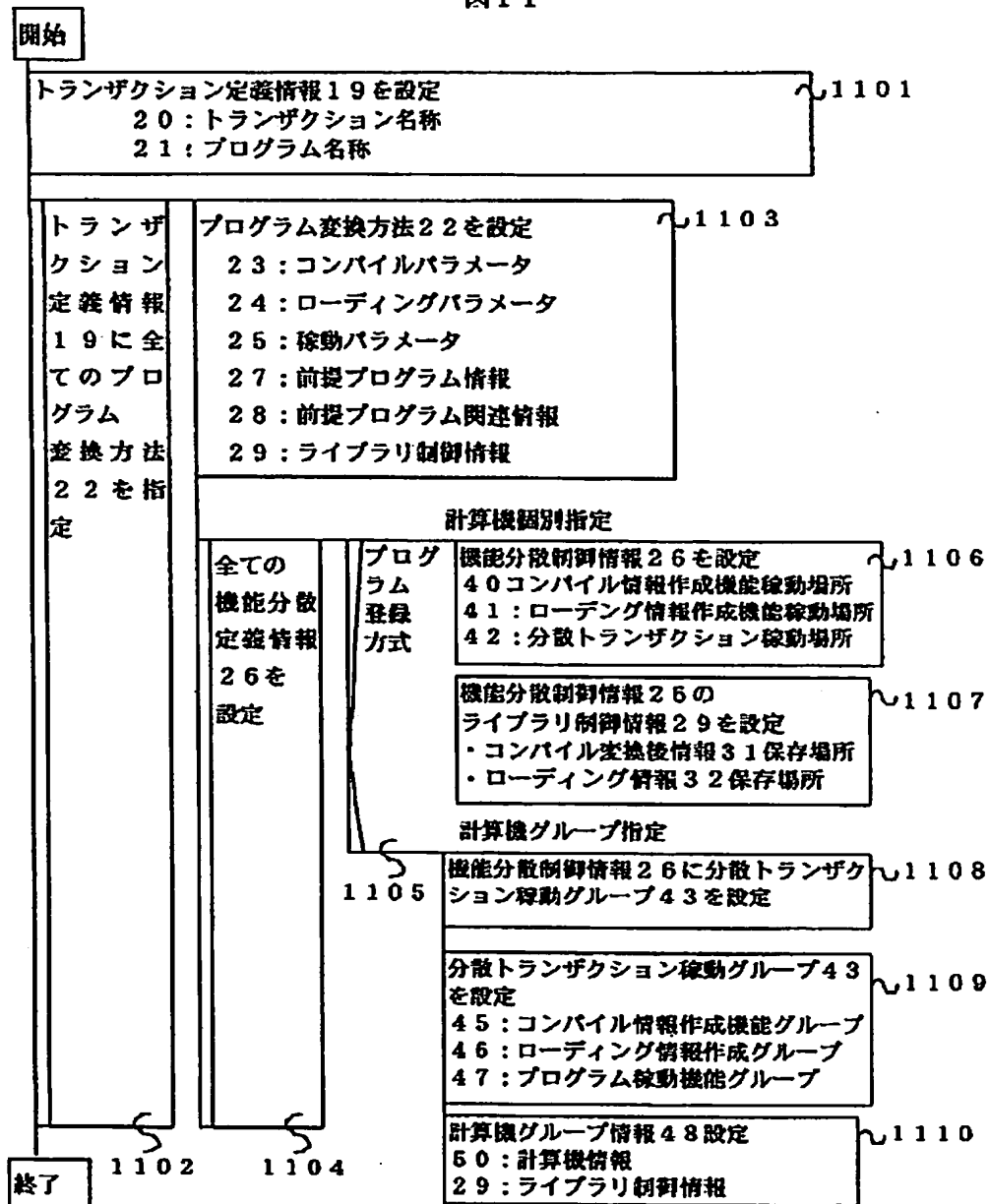


機能分散制御情報26指定項目 (例: 機能提供サーバ18の利用例)

指定情報	指定内容(例)
40 コンパイル情報作成機能稼働場所	計算機C(機能提供サーバ)
41 ローディング情報作成機能稼働場所	計算機C(機能提供サーバ)
42 分散トランザクション稼働場所	計算機B
43 トランザクション稼働グループ	— (設定無し)
28 ライブラリ制御情報 (コンパイル交換情報)	計算機C
29 ライブラリ制御情報 (ローディング情報)	計算機C

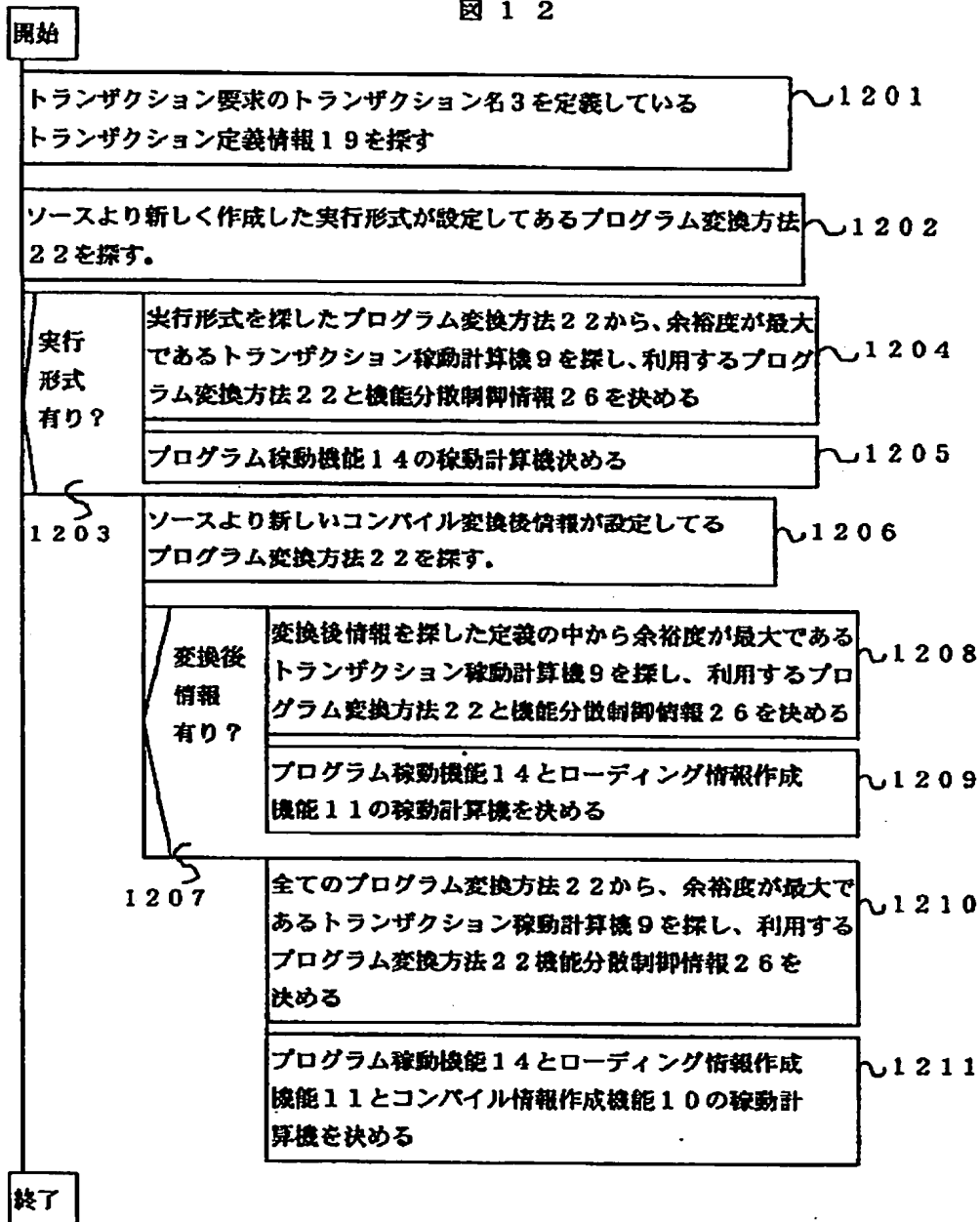
【 図11 】

図11

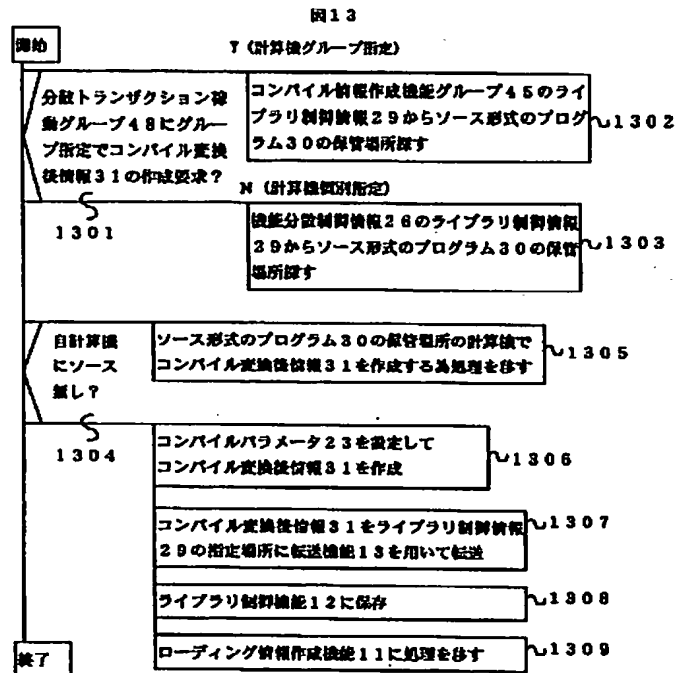


【 図12 】

図 1 2



【 図13 】



【 図14 】

